



1130 Commercial Subroutine Package, Version 2

Application Description

The IBM 1130 Commercial Subroutine Package is for 1130 users with a knowledge of FORTRAN. The routines are intended to give commercial capability to users of IBM 1130 FORTRAN; they are not intended to make FORTRAN a complete commercial language. This manual provides sufficient information to determine the applicability of the package.

Third Edition

This edition, H20-0221-2, which is a major revision of, and obsoletes, H20-0221-1, contains an overall description of added capabilities available in Version 2 of the program.

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

INTRODUCTION

Commercial programming has certain requirements that are not readily available in FORTRAN, a scientifically oriented language — for example, the ability to:

1. Read a record and extract information from it, when the record format is determined by codes within the record.
2. Analyze and make logical decisions on the basis of alphameric information, rather than strictly numeric information.
3. Edit with commercially used characters, thus providing output records with a more finished appearance.

The IBM 1130 Commercial Subroutine Package (APS # 1130-SE-25X), Version 1, enables the FORTRAN user to perform the basic functions of commercial programming. The following commercial capabilities are provided:

1. Floating dollar sign and asterisk check protection
2. Alphameric move and compare operations
3. Ability to read unformatted records
4. Complete input/output character editing with zone punch manipulation

The IBM 1130 Commercial Subroutine Package is modular in design, and Version 1 consists of the following eight subroutines:

1. MOVE — Move a variable amount of data
2. EDIT — Edit data
3. GET — Extract a field from an input area
4. PUT — Place a variable in an output area
5. NCOMP — Compare two variable-length, alphameric fields
6. NZONE — Zone testing and zone modification
7. FILL — Fill a specified area
8. STACK — Select the alternate stacker

Two additional requirements that have been found for commercial programming are the ability to:

1. Perform calculations with very large numbers (greater than 1 billion) and get exact results (to the penny).
2. Perform I/O operations concurrently with other CPU operations.

The IBM 1130 Commercial Subroutine Package, Version 2, enables the FORTRAN user to perform these additional basic functions. The following additional commercial capabilities are provided:

1. Decimal arithmetic operations
2. Overlapped I/O

The IBM 1130 Commercial Subroutine Package, Version 2, is also modular in design and consists of the following subroutines in addition to the subroutines in Version 1:

1. ADD — variable-length decimal add
2. A1DEC — conversion from A1 format to decimal format
3. DECA1 — conversion from decimal format to A1 format
4. DIV — variable-length, decimal divide
5. ICOMP — compare two variable-length, decimal data fields
6. IOND — wait until all I/O operations are finished
7. MPY — variable-length, decimal multiply
8. NSIGN — test a sign, or modify a sign
9. PACK/UNPAC — conversion between A1 and A2 format
10. PRINT/SKIP — overlap the printing, or skip the carriage on the 1132 Printer
11. READ/PUNCH — read or punch a card on the 1442 Card Read Punch
12. SUB — variable-length, decimal subtract
13. TYPED/KEYBD — read from the keyboard, or overlap the typing of a line on the console printer

The IBM 1130 Commercial Subroutine Package is designed for an IBM 1130 Computing System with 8,192 words of core storage, with card or paper tape input/output, and with or without disk. The package is written in FORTRAN and the Assembler Language for the IBM 1130. The total core storage requirement for Version 2 is approximately 4,400 words, including the 1,200-word total for Version 1. However, the actual core required may be less than 4,400 if all additional capabilities are not used.

The remainder of this application description outlines the specifications of the subroutines currently available in Version 1.

SUBROUTINE SPECIFICATIONS

The following definitions will hold for the parameters to each subroutine:

- JCARD — A one-dimensional, integer array defined in a DIMENSION statement
- KCARD — A one-dimensional, integer array defined in a DIMENSION statement
- J — An integer variable, an integer expression, or an integer constant
- JLAST — An integer variable, an integer expression, or an integer constant
- K — An integer variable, an integer expression, or an integer constant
- KLAST — An integer variable, an integer expression, or an integer constant
- N — An integer variable, an integer expression, or an integer constant
- NEWZ — An integer variable, an integer expression, or an integer constant
- NOLDZ — An integer variable
- SHIFT — A floating-point variable, a floating-point expression, or a floating-point constant
- VAR — A floating-point variable, a floating-point expression, or a floating-point constant
- ADJST — A floating-point variable, a floating-point expression, or a floating-point constant
- NCH — An integer variable, an integer expression, or an integer constant

All information, unless otherwise specified, is assumed to be one character per word, in A format (this will be referred to as A1 format).

MOVE

The MOVE subroutine is used as follows:

CALL MOVE(JCARD, J, JLAST, KCARD, K)

The information in the data field starting with JCARD(J) and ending with JCARD(JLAST) is moved, left to right, to the data field starting with KCARD(K).

EDIT

The EDIT subroutine is used as follows:

```
CALL EDIT(JCARD, J, JLAST, KCARD, K, KLAST)
```

The edit mask is contained in the field starting with KCARD(K) and ending with KCARD(KLAST). It is composed of the following control characters:

<u>Character</u>	<u>Function</u>
b (blank)	Replace with a source character
0 (zero)	Used in place of a blank; indicates the rightmost limit of zero suppression
* (asterisk)	Indicates the rightmost limit of asterisk check protection
\$ (dollar sign)	Indicates floating dollar sign
(decimal point)	Placed in result field
, (comma)	Placed in result field; may be zero suppressed
- (minus)	If source is negative, placed in result; otherwise, replaced by blank
CR (credit)	Same as minus

The edit mask is destroyed after editing is complete. The source field, starting with JCARD(J) and ending with JCARD(JLAST), is edited into the edit field under control of the edit mask.

GET

The GET routine is a FUNCTION subprogram and, as such, may be used as any standard FORTRAN subprogram (for example, SIN (), SQRT ()).

It is used as follows:

```
A = GET(JCARD, J, JLAST, SHIFT)
```

The data field starting with JCARD(J) and ending with JCARD(JLAST) is converted to a floating-point number, with decimal point at the right end. The value SHIFT multiplies this floating-point number, thus placing the decimal point. SHIFT has the value 10^{-d} , where d is the number of decimal places. The sign of the field is assumed to be over the units position. (Further information may be found in the NZONE description.)

The GET routine is used to extract a field from an input area. Therefore, the GET routine supplies the ability to "read" unformatted records.

The limits on the size of numbers are from -100,000,000.0 to +100,000,000.0.

Minimum	-838,860.8	-214,748,364.8
Maximum	838,860.7	214,748,364.7

PUT

The PUT subroutine is used as follows:

```
CALL PUT(JCARD, J, JLAST, VAR, ADJST, N)
```

The floating-point number, VAR, is first half-adjusted by the parameter ADJST and then N digits are truncated. It is then converted to EBCDIC character codes, one per word, in A1 format. The result is placed in the data field starting with JCARD(J) and ending with JCARD(JLAST), right-justified.

NCOMP

The NCOMP routine is a FUNCTION subprogram. It is used as follows:

```
IF (NCOMP(JCARD, J, JLAST, KCARD, K)) 1, 2, 3
```

The data field starting with JCARD(J) and ending with JCARD(JLAST) is compared to the data field starting with KCARD(K). If the field at JCARD is greater than the field at KCARD, control goes to statement 3; if the field at JCARD is less than the field at KCARD, control goes to statement 1; if the fields are equal, control goes to statement 2. The collating sequence, in ascending order, is A-Z, 0-9, special characters. The compare is from left to right and is terminated by (1) corresponding characters are not equal, or (2) the character at JCARD(JLAST) has been compared.

NZONE

The NZONE subroutine is used as follows:

```
CALL NZONE(JCARD, J, NEWZ, NOLDZ)
```

The character at JCARD(J), in A1 format, is interrogated. The following tables define the values of NOLDZ and NEWZ:

<u>If the character is</u>	<u>then NOLDZ is set to</u>
A-I	1
J-R	2
S-Z	3
0-9	4
Special	greater than 4

<u>If NEWZ is</u>	<u>then the zone is changed to</u>
1	12— zone
2	11—zone
3	0—zone
4	no—zone
greater than 4	no change

If the character is originally a special character, it is not changed.

There is one special case that should be mentioned. An 11-0 Hollerith combination is not a legal FORTRAN character. An input card column containing an 11-zone over a zero will be converted by FORTRAN to a blank, and for output, an EBCDIC 11-zone over a zero will be punched as a blank. This problem can be avoided by using the convention that, externally, an 11-zone or - will represent a zero with an 11-zone. On input, if the zone is removed by the subroutine, an 11-zone is converted to a zero. On output, if an 11-zone is placed over a zero, the subroutine will delete the zero.

FILL

The FILL subroutine is used as follows:

```
CALL FILL(JCARD, J, JLAST, NCH)
```

The data field starting with JCARD(J) and ending with JCARD(JLAST) is filled with the EBCDIC character corresponding to the value of NCH.

STACK

The STACK subroutine is used as follows:

```
CALL STACK
```

Only the next card to pass through the punch station is selected into the alternate stacker of the IBM 1442 Card Read Punch.

MACHINE AND SYSTEM CONFIGURATION

The IBM 1130 Commercial Subroutine Package is designed for execution on an IBM 1130, Model 1B or 2B, with a 1442 Card Read Punch, Model 6 or 7. In addition, the Console Printer, 1134 Paper Tape Reader, 1055 Paper Tape Punch, and 1132 Printer are supported. For compilation and assembly only, the minimum 1130 FORTRAN card system requirements are sufficient. The package is written in FORTRAN and in the Assembler Language for the IBM 1130.

BIBLIOGRAPHY

IBM 1130 Computing System, Principles of Operation, Systems Reference Library (A26-5881).

IBM 1130 Assembler Language, Systems Reference Library (C26-5927).

IBM 1130 Subroutine Library, Systems Reference Library (C26-5929).

IBM 1130 FORTRAN Language, Systems Reference Library (C26-5933).

READER'S COMMENT FORM

1130 Commercial Subroutine Package, Version 2

H20-0221-2

Application Description

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

—
fold

—
fold

—
fold

—
fold

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, STAPLE AND MAIL.

YOUR COMMENTS PLEASE...

Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material.

Please note that requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or the IBM branch office serving your locality.

fold

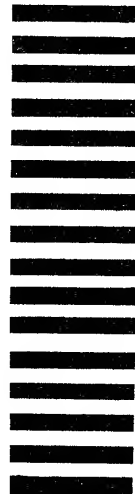
fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

IBM Corporation
112 East Post Road
White Plains, N. Y. 10601



Attention: Technical Publications

fold

fold

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]



International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
(USA Only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)